

EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Timothy Sullivan on 09/09/2008.

Claims 1 and 11-20 of the application has been amended as follows:

1 (currently amended): A computer-implemented method for binding data in a user interface (UI) script, comprising:

generating a tree structure that corresponds to the UI script;

automatically determining whether the tree structure includes cloned templates that were previously grafted to the tree structure and automatically removing any cloned templates that were previously grafted to the tree such that templates that already include data from an external data source are removed from the tree, ~~such that wherein~~ the tree structure is in a state where data binding has yet to occur; ~~wherein and~~ the cloned templates are removed before ~~accessing~~ cloning a reference template;

accessing the reference template;

cloning the reference template to create a cloned reference template while maintaining the reference template such that the reference template is available for subsequent iterations of binding data;

inserting the data into the cloned reference template;

grafting the cloned reference template into the tree structure after the data has been inserted into the cloned reference template; and

displaying a UI output according to the tree structure, whereby the UI output is dynamically updated with the data.

11 (currently amended): A computer-readable storage medium that includes computer-executable instructions for binding data to a user interface (UI) script, comprising:

generating a tree structure that corresponds to the UI script;

automatically determining whether the tree structure includes cloned templates that were previously grafted to the tree structure and automatically removing any cloned templates that were previously grafted to the tree such that templates that already include bound data from an external data source are removed from the tree, such that wherein the tree structure is in a state where data binding has yet to occur; wherein and the cloned templates are removed before accessing cloning a reference template;

cloning the reference template to create a cloned reference template while maintaining the reference template ~~for later use~~ such that the reference template is

available for subsequent iterations of binding data; wherein a portion of the reference template is associated with a portion of the UI script that includes a placeholder for data;

replacing the placeholder in the cloned reference template with the data;

grafting the cloned reference template into the tree structure; and

displaying a UI output according to the tree structure, whereby the UI output is dynamically updated with the data.

12 (currently amended): The computer-~~readable~~ storage medium of claim 11, wherein the tree structure and the UI script are logically equivalent.

13 (currently amended): The computer-~~readable~~ storage medium of claim 11, wherein cloning the reference template further comprises determining which portions of the tree structure correspond to a specified tag of the UI script.

14 (currently amended): The computer-~~readable~~ storage medium of claim 13, wherein the specified tag has an associated attribute for retrieving multiple records for display in the UI output as a list of records.

15 (currently amended): The computer-~~readable~~ storage medium of claim 13, wherein the specified tag has an associated attribute that specifies an interval for refreshing the data.

16 (currently amended): The computer-~~readable~~ storage medium of claim 11, wherein replacing the placeholder further comprises retrieving the data from an external data source.

17 (currently amended): The computer-~~readable~~ storage medium of claim 16, wherein replacing the placeholder further comprises passing a uniform resource locator (URL) that identifies a location of the data to a communication library.

18 (currently amended): The computer-~~readable~~ storage medium of claim 11, wherein a state is associated with each portion of the tree structure in which a placeholder is present.

19 (currently amended): The computer-~~readable~~ storage medium of claim 18, wherein a first component is displayed in the UI when the state corresponds to a first state, and a second component is displayed in the UI when the state corresponds to a second state.

20 (currently amended): A system for binding data to a user interface (UI) script, comprising:

memory;

a target user interface device that includes a first application that is configured to:
generate a tree structure that corresponds to the UI script;

automatically determining whether the tree structure includes cloned templates that were previously grafted to the tree structure and automatically removing any cloned templates that were previously grafted to the tree such that templates that already include data from a separate data source are removed from the tree; wherein the tree structure is in a state where data binding has yet to occur and the previously grafted cloned templates are removed before cloning a reference template;

cloning the reference template to create a cloned reference template while maintaining the reference template such that the reference template is available for subsequent iterations of binding data;

inserting the data into the cloned reference template;

grafting the cloned reference template into the tree structure after the data has been inserted; and

displaying a UI output according to the tree structure, whereby the UI output is dynamically updated with the data.

REASONS FOR ALLOWANCE

1. Claims 1-25 are allowed.
2. The following is a statement of reasons for the indication of allowable subject matter:

In interpreting the claims in light of the specification and applicant's arguments, the Examiner finds the claimed invention is patentably distinct from the prior art of record.

The prior art of record includes Burkett et al (Burkett), US Patent No. 6,635,089, Pik et al (Pik), US Patent Application Publication No. US 2004/00230906, Messina, US Patent No. 5,634,128, and Boehme et al (Boehme), US Patent No. 6,578,192.

Burkett discloses that documents encoded according to the XML notation can be marked up to indicate that the corresponding DOM tree is to be updated dynamically to reflect changing information (Abstract, Figures 4A&4B, 5A-5E, col. 9, line 44 - col. 11, line 14). Burkett further discloses the DOM tree is generated and the DOM API enables application program (automatically) to access a tree-oriented abstraction of a document, and to manipulate document structure and contents (that is, by changing, deleting, and/or adding elements), i.e., when a query-results node is present, the parsed query result is inserted into the DOM tree as a subtree beneath that node, replacing any subtree that previously existed (col. 18, lines 34-38). Burkett further discloses The DOM tree enables navigating the structure of the document (col. 1, lines 35-58). Burkett further discloses dynamically constructed nodes 421a, 422a and 423 a in Figure

4E have been substituted for nodes 415, 416 and its subtree comprised of nodes 417, 418a-b and 419, and this implies inserting the data into a portion of the tree structure and grafting the portion of the tree into the tree structure. Since Burkett discloses substitute one portion of the tree with different data, the step of cloning a portion of the tree structure should be inherent in the above substitute step.

Pik discloses storing the scripting object can be involve copying a scripting object reference to the shared memory to cause retention of the scripting object by the client during document loading, then cloning the stored scripting object to the newly loaded document can be performed and can involve creating a new scripting object in the new document and copying data from the stored scripting object to the new scripting object (Abstract, Figures 2-3). Pik also discloses in pages 3-4, paragraphs [0028]-[0035] that the events are user triggered events that change the state of a scripting object loaded in the page, and during the document reload, the stored scripting object can be cloned to the reloaded document, creating a cloned scripting object in a reloaded document memory space. Moreover, a reference to the scripting object of the loaded document can be copied to a shared memory and this causes retention of the scripting object during the document reload (page 3, paragraph [0032]). Figures 4-7 and page 4, paragraphs [0036]-[0039] of Pik reference provide a simple point and click user interface to a navigation hierarchy.

Messina discloses controlling the access to objects stored in a data processing system wherein the objects are hierarchically structured (tree dom) and each object being either locally available or retrievable from an external unit (col. 2, lines 14-25).

Messina further discloses all of the objects contain information about the existence of dependent objects (a counter is associated with each object indicating whether there are objects dependent on the object) (col. 4, lines 1-6). Messina further discloses that through the dependent object counter, a processing program can determine if any locally available object has a dependent object, then appending a dependent object to a locally available object when at least one related dependent object is non-locally available (col. 4, lines 10-15). Messina further discloses that a place-holder graphic symbol (template) is displayed to represent all of the non-locally available dependent objects, and when all of the non-locally available objects are retrieved, the place-holder graphic symbol (template) is removed from the display (tree) and the dependent object is deleted from the tree in the memory (col. 2, lines 34-48). Thus, these steps imply automatically removing any templates that were previously grafted to the tree such that templates that already include data from an external data source are removed from the tree.

Boehme discloses a document is loaded having script components which are identified and delineated and then passed to an interpreter to returns an object corresponding to each script component (Abstract and col. 3, line 59 – col. 4, line 38, and col. 7, lines 11-56).

Claim 1 is allowed because the prior art of record does not expressly disclose alone or in combination automatically determining whether the tree structure includes cloned templates that were previously grafted to the tree structure and automatically

removing any cloned templates that were previously grafted to the tree such that templates that already include data from an external data source are removed from the tree, wherein the tree structure is in a state where data binding has yet to occur and the cloned templates are removed before cloning a reference template; and cloning the reference template to create a cloned reference template while maintaining the reference template such that the reference template is available for subsequent iterations of binding data.

3. Dependent claim 2-10 further limit independent claim 1. Claims 11-25 are allowed as well for the same reasons set forth for claims 1-10.

4. Any comments considered necessary by applicant must be submitted no later than the payment of the Issue Fee and, to avoid processing delays, should preferably accompany the Issue Fee. Such submission should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chau Nguyen whose telephone number is (571) 272-4092. The Examiner can normally be reached on Monday-Friday from 8:30 am to 5:30 pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Doug Hutton, can be reached at (571) 272-4137.

The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306. On July 15, 2005, the Central Facsimile (FAX) Number will change from 703-872-9306 to 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chau Nguyen
Patent Examiner
Art Unit 2176

/Rachna S Desai/
Primary Examiner, Art Unit 2176